Computers Math. Applic. Vol. 20, No. 4–6, pp. 247–267, 1990
Printed in Great Britain

0097-4943/90 $3.00 + 0.00
Pergamon Press plc

THE CELLULAR DEVICE MACHINE: POINT OF DEPARTURE FOR LARGE-SCALE SIMULATIONS OF COMPLEX BIOLOGICAL SYSTEMS

H. B. SIEBURG

Department of Psychiatry, University of California, San Diego, Mail Stop M-003-H, La Jolla, CA 92093, U.S.A.

Abstract—The cellular device machine (CDM) is the computer implementation of a general mathematical modeling approach, called the nested automaton, suitable for large-scale simulations of complex systems. Due to this method, a large body of experimental data representing the structure and behavior of a complex system can be compressed to allow its description in terms of objects and communication between them. In its applications, this object-oriented and event-driven description can be easily extended and modified to allow the mathematical analysis of fundamental system properties and system dysfunctions prior to experimental analysis. The present paper focuses on the immune system as an example to illustrate the capabilities of the nested automaton approach to (1) substantially reduce the complexity arising from the building structure of large and heterogeneous systems and (2) show the capacity of the CDM to exhibit and explain phenomena of self-organization in such systems. The latter is explicated in brief applications concerning the regulation of growth in the immune system and the pathogenesis of acquired immune deficiency syndrome (AIDS).

INTRODUCTION

Modern computer simulation and mathematical modeling comprise important investigative techniques which provide the means for investigating systems whose behavior derives from the interactions between large populations of discrete, structurally similar, but spatially and informationally heterogeneous objects. Such systems are called complex.

Many important behavioral aspects of biological systems such as spontaneous self-organization, nonlinearity, openness, redundancy and hysteresis [1] derive from complexity. Most current approaches to the computer simulation of biological systems are based on differential equations or networks. Differential equation models describe continuously evolving systems with a small number of continuous degrees of freedom. Network modeling provides for interconnected system compartments or network nodes. These connections are integral part of the model definition and establish a static linkage for the duration of a simulation. Both approaches are limiting when used to describe complex systems which involve many different event-driven processes occurring simultaneously.

The recent availability of high-performance (parallel-) computer architectures and dedicated workstations suggested to us that the resurrection of classical artificial intelligence (AI) techniques [2] and their conjunction with "new wave" mathematical methodologies might open novel approaches for the modeling of complex biological systems. Here, "object-oriented programming", "finite automata" and "cellular automata" seemed particularly interesting. The AI technique of "object-oriented programming" [3] describes a system in terms of objects and dynamic linkages (communication) between them. Links (messages) between objects appear and disappear dynamically as functions of the states of the objects at various points in time. The mathematical theory of "finite automata" [4] can be used to describe objects in terms of developmental stages again connected by dynamic linkages. Transitions between stages are accomplished depending on the recognition of distinctive information. Finally, abstract parallel computers or "cellular automata" [5] provide a space–time arena where object migration can be conveniently simulated and system evolution can be studied.

The present paper describes how these state-of-the-art techniques can be integrated to construct a special-purpose computer, called the cellular device machine (CDM) [6, 7], suitable for the mathematical exploration of complex biological systems (see Fig. 1). The central paradigm of this novel mathematical modeling methodology is that complex behavior in large and heterogeneous

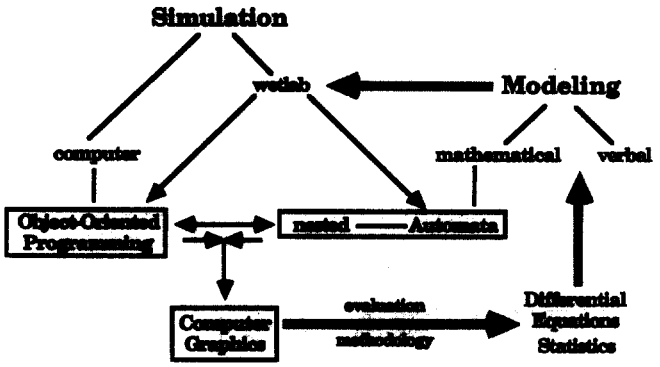


Fig. 1. Major methodological building-blocks (rectangles) of the cellular device machine concept. The choice of the indicated methodologies (rectangles) was motivated by the observation of experimental approaches (wetlab simulation). The linkage between the (theoretically equivalent) object-oriented programming style and the novel nested automaton appraoch is provided by an interactive graphics interface. A novel evaluation procedure is required to interpret emerging simulation patterns before they can be returned to the experimental system via mathematical and verbal models (shaded arrows).

biological systems arises from event-driven interactions between simple objects. These objects are determined by three independent parameters, namely *structure*, spatial *location* and the capacity to recognize and process *information*.

The immune system is used as an example to illustrate this paradigm. Biologically, this system is composed of large populations of cells (objects) which are functionally interrelated by a large number of cooperative processes. Many of these processes are innitiated by the specific recognition of antigen (event-driven) and evolve into complex patterns of cellular signalling through interdependent secreting and binding of immune system proteins (messages). It has become clear that immune system complexity requires theoretical models as essential elements of experimental and clinical research [8]. It is now also evident that novel mathematical and computational methods have to be sought to replace traditional, now insufficient

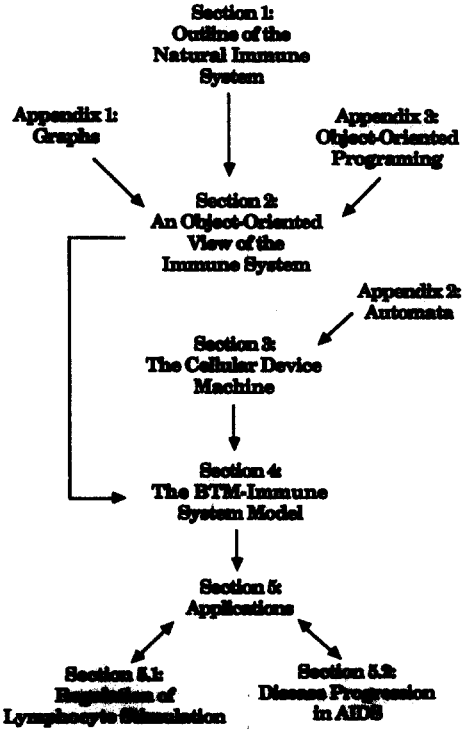


Fig. 2. Paper organization overview chart.

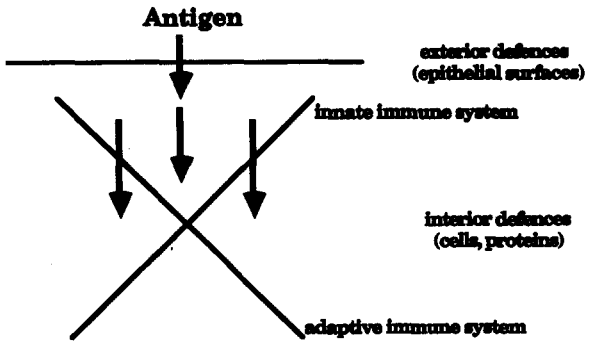


Fig. 3. A foreign intruder, having breached a host's exterior defences, encounters the cellular and humoral components of the functionally integrated innate and adaptive compartments.

approaches [9]. The CDM, when applied to the immune system, constitutes such a novel methodology.

The paper is organized as follows (see Fig. 2): The first section outlines the biology of the immune system. In the second section, graph theory (Appendix A) is applied to derive an object-oriented, event-driven mathematical description of the system. In the third section the actual simulation tools, i.e. cellular automata (Appendix B), the novel nested automata [6, 7] and object-oriented programming (Appendix C), are combined to derive the main parts of the CDM. In the fourth section it is shown how an immune system consisting of mature B cells, T helper cells, T suppressor cells and macrophages can be implemented on the CDM (the BTM-machine). The concluding fifth section summarizes results obtained in on-going projects where the BTM-machine is applied to the regulation of antigen-specific responses and the modeling of infectious immune system diseases.

The paper covers a broad area, and includes many mathematical details and "how to" suggestions, which worked for us. It is, however, intended as a survey and progress report rather than a rigorous mathematical treatment.

OUTLINE OF THE NATURAL IMMUNE SYSTEM

Against the intrusion by foreign pathogens, generically defined as antigens, an individual is usually well-protected by a system of exterior defences such as the skin, acid in the stomach, mucus etc. Once these exterior defences are breached, the intruder encounters the host's interior defence system, i.e. the cells and proteins of the immune system (see Fig. 3).

One can roughly distinguish two compartments. Namely, the innate immune system, which recognizes antigen unspecifically, and the adaptive immune system, which recognizes antigen

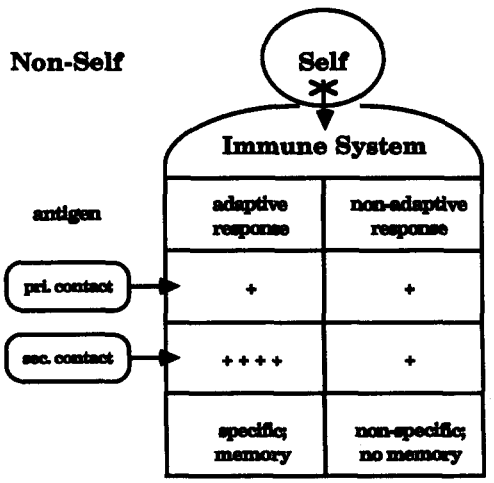


Fig. 4. The immune system discriminates self from nonself, and reacts against nonself molecules (antigens). The characteristics of an adaptive immune response are specificity and memory.

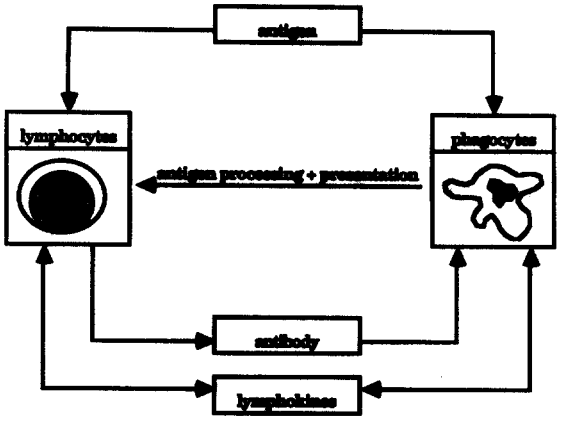


Fig. 5. Functional integration of innate and adaptive immunity.

specifically by virtue of a large variety of adaptor molecules, called antigen receptors. The most commonly known molecules of this kind are the antibodies. Should the infectious agent persist or challenge the host a second time, the adaptive compartment is able to mount a largely enhanced secondary response due to a process called memory. In comparison, the secondary response mounted by the innate immune system, which is unspecific, nonadaptive and has no memory, is the same as its primary response (see Fig. 4).

Both compartments make a crucial distinction prior to launching a response. Namely, they distinguish between self and the outside world or nonself. This distinction is learned and is effectual through the system's antigen receptor variety called the repertoire. The repertoire is formed prior to any actual encounter with antigen, i.e. the recognition mechanism of the immune system relies on event anticipation, and is variable under germline and somatic mutation.

The innate and adaptive compartments are not independent units. They are functionally integrated by a large number of cooperative processes between their cellular components. Many of these processes are initiated by antigen and evolve into complex patterns of cellular signalling through interdependent secreting and binding of immune system proteins (see Fig. 5). In the example shown here, the phagocytes, which are part of the innate immune system, process and present antigen to stimulate lymphocytes, which are part of the adaptive immune system. Upon stimulation and subsequent exchange of signaling proteins called lymphokines, lymphocytes undergo a number of differentiation and proliferation steps which lead—in the case of B lymphocytes—to large clones of antibody-producing or plasma cells. The availability of large quantities of specific antibody in turn facilitates antigen-binding to phagocytes.

All cells of the immune system derive from the pluripotent stem cells which reside in the bone marrow. Therefore, we can call the bone marrow the source cluster of the immune system. In Fig. 6 below, only B cells, T cells and macrophages are indicated due to their relevance in the applications presented later in this paper.

Coming from the bone marrow, immune system cells undergo a series of differentiation steps in specific microenvironments, e.g. T cells in the thymus, before they enter the blood circulation as mature cells. From there, via the lymphatic system, they migrate into the secondary lymphoid tissues such as the lymph nodes. Here they remain, organized in distinctive patterns, before they recirculate into the system. Therefore, other than of cells and proteins, the immune system consists of fluids for transport, i.e. blood and lymph, and of organs and tissues supporting cell development and/or antigen-trapping. Antigen-trapping in these highly organized cellular clusters provide the threshold concentrations and cell-packing favorable to a successfully mounted immune response.

In summary, we are confronted with an anticipatory system, which:

—is very large, e.g. in humans the number of cells is thought to be in the order of 10^{12};
—is heterogeneous, i.e. one finds many different cell types, each of which follows its own distinctive differentiation pathway;

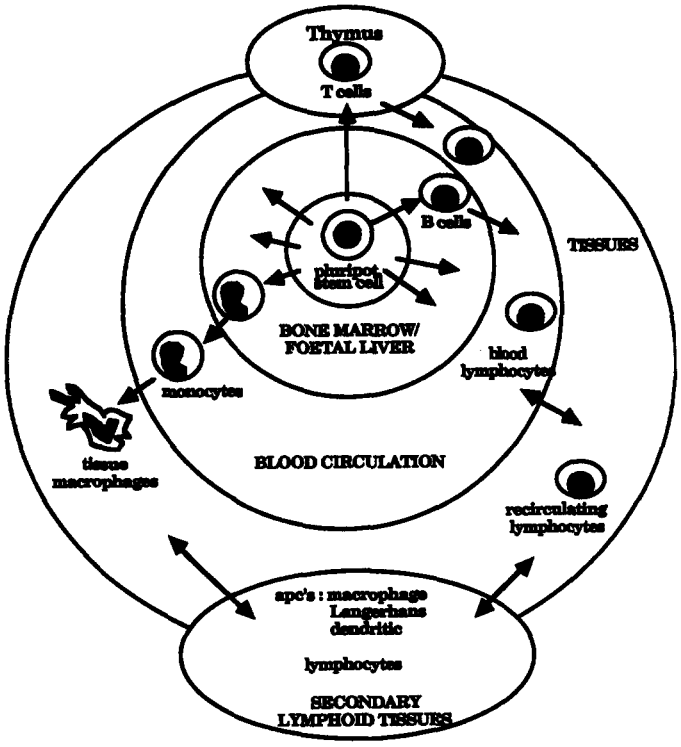


Fig. 6. Origin of immune system cells.

—has a large variety of specificities for recognition, e.g. in mammals the number of cell clones with different specificity is 10^6–10^8, each of which can consist of as little as 1 cell and as many as 10^6 cells.

The system's behavior is determined by the

—interaction of many components, which
—migrate between
—highly organized cellular clusters, which support
—cell development and antigen-trapping.

(See Fig. 7.)

AN OBJECT-ORIENTED VIEW OF THE IMMUNE SYSTEM

By looking thoroughly into the biology, we found that crucial simplifications can be administered in this seemingly overwhelming list of modeling problems. The resulting object-oriented view of the immune system is presented in an integrated scheme of biological observations and mathematical conclusions. The mathematical notion of "graph" which is used in this scheme is defined and expanded upon for the reader's convenience in Appendix A.

Observation 1

The immune system is a self-organized system whose components, the cells, are the producers, consumers and carriers of information. Cells interact in response to events such as the specific recognition of antigen and are capable of proliferation.

Conclusion 1 ("kinetisity")

The immune system can be defined as a virtually connected, expandible set O of objects. The connections are dynamically allocatable and event driven, i.e. links (messages) between objects appear and disappear as functions of the states of the objects and the input received.

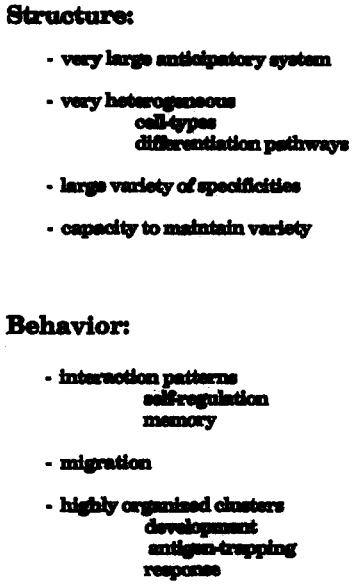


Fig. 7. Immune system structure/behavior summary chart.

Observation 2

Each of the large number of cells in the immune system undergoes a small number of differentiation steps during a finite lifetime. Differentiation steps are executed sequentially following an ordered scheme. From a certain differentiation stage on, the fate of a cell depends on its history and the interaction with other cells.

Conclusion 2

O can be reduced to a set of directed graphs g whose nodes relate to cellular differentiation stages. For every g there exists an integer-valued function $E(g, T)$—a biological clock—which slowly decreases with respect to a universal time scale T and increases depending on transitions between nodes of g.

Observation 3

The transition from one differentiation stage to another depends on the availability of very distinctive information at the right time. In other words, the differentiation process requires inertia, and receptors in anticipation of particular kinds of information.

Conclusion 3 ("continuous availability")

The nodes of any $g \in O$ are capable of communication executed on the basis of a characteristic set of signals. In the language of graph theory, all g are therefore labeled and colored.

Observation 4

All members of a particular cell type are identical with respect to their developmental capacity and their ability to interact with members of other cell types ("structural equivalence"). They are different with respect to their ability to recognize information ("informational diversity").

Conclusion 4 ("modularity")

For every cell-type X there exists a directed, colored and labeled graph g_X such that

$$X = \langle g_X \rangle,$$

i.e. X is representable by a unique structural generator. Therefore, the set of immune system objects is characterizable by a small set of generators

$$O = \{\langle g_{X_1} \rangle, \ldots, \langle g_{X_q} \rangle\},$$

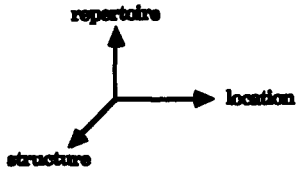


Fig. 8. The dimensions of the abstract dynamical system created in the CDM. Each object represented in CDM is determined by a linked set of developmental stages (structure), its position in the physical space–time universe (location) and, its capacity to recognize information (repertoire). The latter, combined with an object's structure and location, will determine its capacity to process and respond to any particular message.

where q denotes the number of different cell types in the immune system. The interaction between cell types can be described in terms of nested graphs Γ, whose node sets $N(\Gamma)$ are subsets of O and whose connections are event dependent.

In summary, given the data compression achieved from these conclusions, a complete computer model of an immune system envolves one kind of (hierarchically organized) objects. Such a model is realistic in the sense that important observable immune system properties, such as self–nonself discrimination, learning, memory, regulation etc., can deduce from the object structure thus provided. Also, as will be seen, the model is predictive. Specifically, the object structure comprises: (1) perceptive objects, which later will be called perceptors, each of which biologically relates to a homogeneous receptor population expressed at a fixed point in time and a receptor population expressed as a consequence of a ligand recognition process involving the first population; (2) perceptor groups, later called cell devices, each consisting of an interconnected finite set of perceptors; and (3) device groups, later called cell modules, each consisting of a finite set of cell devices and a set of attributes, which define their interactive potential.

THE CDM

In this section we survey the components of our main simulation tool, the CDM. The CDM is a special-purpose computer designed for the mathematical exploration of biological systems. The central paradigm of the CDM is that the complex behavior of large and heterogeneous biological systems originates from the dynamics of interaction between simple objects. Under this paradigm the abstract object space is determined by three independent parameters, namely *structure*, spatial *location* and the capacity to recognize and process *information* (Fig. 8).

The CDM has four basic parts (see Fig. 9):

(1) The BODY, which is a two-dimensional cellular automaton represented by an interface in user-definable planar topologies. The default topology is a rectangular grid of sites.

A site can hold at most one object and a number of signals at a time. Starting from an initial distribution, objects are born into the "body" through proliferation. They evolve according to events and their lifetime is dependent on the encounter and successful recognition of system internal or external messages. The cellular automaton transition rule organizes the input to and the output from each site with respect to a set of nearest neighbor sites, and further allows us to spatially dislocate objects and messages (Fig. 10).

Mathematically, the "body" represents a four-dimensional, spatially and temporally homogeneous space with discrete space–time decomposition where information is generated, processed, propagated etc. locally.

(2) The OBJECT STACK, which holds all objects created by the user.

Objects are created using a nesting procedure starting from simple 2-state automata, called perceptors, whose initial states are capable of specifically recognizing bit-patterns called messages. Upon recognition, a perceptor will "differentiate" to form the new perceptor state. A biological example for a perceptor is given by the genetically controlled mechanism connecting a homogeneous receptor population phenotypically expressed on the surface of a cell with one subsequently expressed as a consequence of successfully accomplished ligand recognition (Fig. 11).

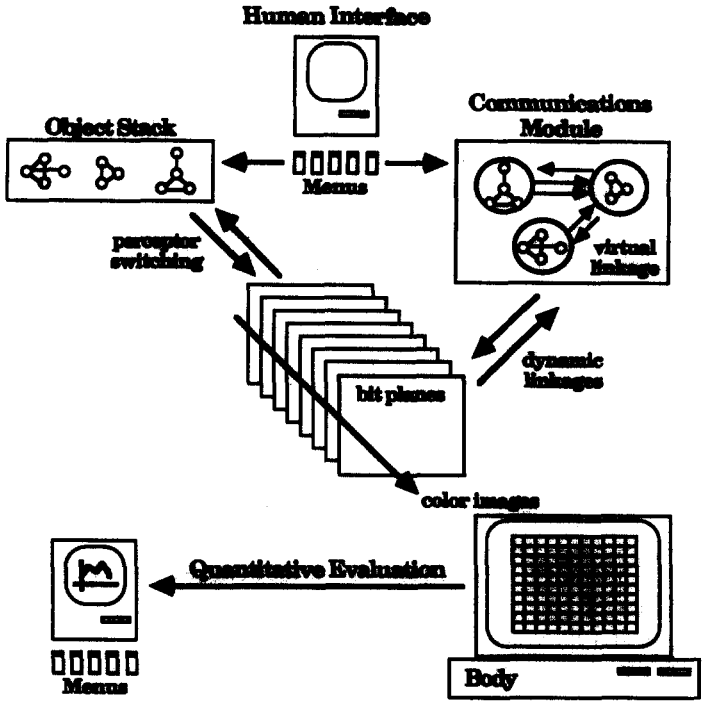


Fig. 9. The CDM consists of three specialized processors. The first harbors the menu-driven human interface, which allows for the definition of objects, virtual messages between them, and real-time interference at this level during a simulation process. In the second, a bitplane is allocated for a population of each object type. The position of an object in its bitplane can be used to refer to both its space–time location and a table of updatable attributes such as specificity, thresholds and eigenlife. The accelerated processing of bitmaps and their cumulative display in color-images is a key-feature of the second processor. These images are transmitted to the menu-driven third processor which uses specially developed high-speed algorithms for quantitative image evaluations.

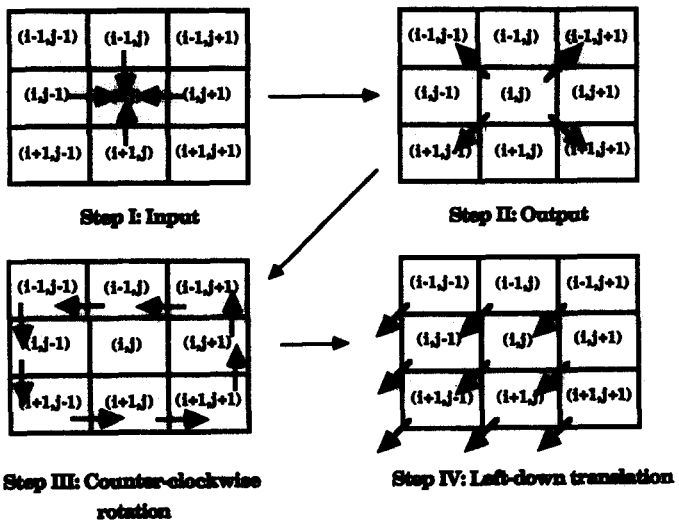


Fig. 10. An algorithm for object parameter changes. For simplicity, the following description uses a rectangular grid of sites with periodic boundary conditions. Under these conditions, the migration of objects located in the sites can be mathematically thought of as taking place on a torus. The algorithm for changing the characteristic parameters (state, eigenlife, receptor populations, location) of an object located in the center site (i, j), consists of four steps: Step I—(i, j) receives messages from four nearest neighbors, processes these messages; and Step II—responds into its remaining four neighbors; Step III—messages perform a one-step counter-clockwise rotation and Step IV—objects and messages perform a one-step "left-down" translation. Steps III and IV are one possibility for implementing migration in the CDM. This is an example of a simple method for realistically considering message-halflives namely by attributing a limit for the number of rotations each message-type can perform.

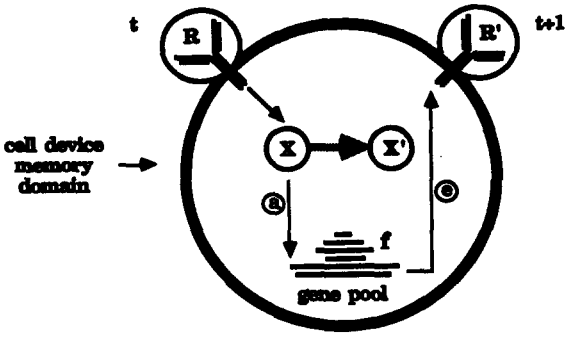


Fig. 11. Perceptor differentiation mechanism. At time t, a cell device whose perceptor $\{X, X'\}$ is active, is assumed to have expressed a receptor population of specificity R. R is represented by a string of bits, e.g. $R = 1110101000011100$ is a valid 16 bit expression. Upon successful recognition of a signal Y (also a bitstring), X will be the transition to the new interior state X' and activate ⓐ the gene pool to express ⓒ a new receptor population of specificity R' at time $t + 1$. The transition is executed by a (set of) function(s) f, capable of manipulating bitstrings (Boolean functions). The notion "cell device memory domain" recalls that the object-oriented implementation provides a "private sector" of computer memory per device.

Connecting a number of perceptors establishes a finite differentiation pattern or cell device (see Fig. 12). Biologically, a cell device encodes a cell as a finite pattern of developmental stages connected by feed-forward transition pathways, where each transition depends on the availability of distinctive information.

In summary, the "body" and the "object stack" together uniquely determine each object in terms of three independent dynamical properties (or dimensions), namely spatial displacement, structure (as given by a differentiation pattern) and the capacity to interact (recognize and transmit information) with other objects in the environment. These properties are efficiently accommodated by a single mathematical structure, the nested automaton. By definition, a nested automaton is a finite automaton (Appendix B),

$$A_1=(\Sigma, \Phi, \Delta, \lambda, \delta),$$

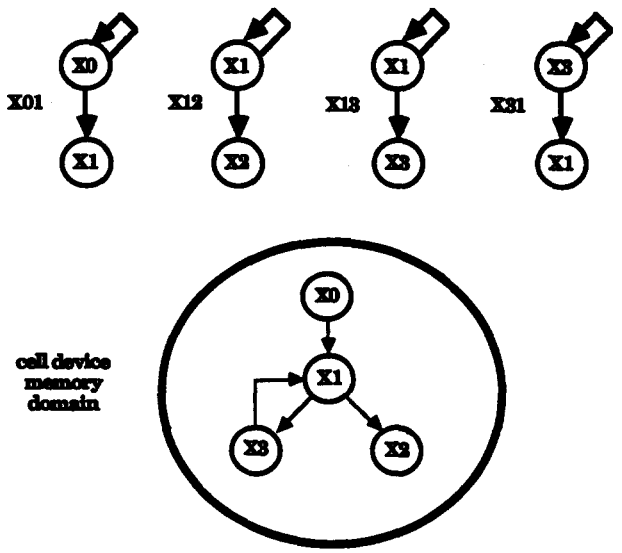


Fig. 12. Cell device object structure. A cell device X is considered whose state set consists of four elements $\{X0, X1, X2, X3\}$. These are thought to be grouped in a pattern of linked perceptors denoted $X01 = \{X0, X1\}, X12 = \{X1, X2\}, X13 = \{X1, X3\}, X31 = \{X3, X1\}$. The individual perceptors can only be active one at a time and thus a differentiating perceptor can subsequently activate only the following one. In the case of bifurcations deterministic or probabilistic rules may be applied to guide the decision process.

whose (finite) state set Φ consists of other finite automata, i.e.

$$\Phi = \{(\Sigma_i, \Phi_i, \Delta_i, \lambda_i, \delta_i): i \text{ in a finite index set } I\}.$$

The input and output sets Σ and Δ, and the transition functions λ and δ, resp., are given by

$$\Sigma = \cup\Sigma_i, \quad \Delta = \cup\Delta_i, \quad \lambda|_{\Sigma_i \times \Phi_i} = \lambda_i, \quad \delta|_{\Sigma_i \times \Phi_i} = \delta_i.$$

The cardinality $d := \#I$ of I indicates the nesting depth. To develop a comprehensive computer model for an immune system we found $d = 3$ to be sufficient.

(3) The COMMUNICATION MODULE, which holds the set of attributes defining the interactive potential of all elements in the "object stack".

In the module, all objects, as represented by their respective cell devices, are joined by dynamically allocatable communication linkages according to a "type" classification of the messages available to the system. Therefore, it is capable of evaluating the information contents of any given neighborhood template in the "body". A particular example for a "communication module" will be shown in the following section (cf. Fig. 13).

By default, the "module" accommodates two major message evaluation tools. One is a recognition mechanism, which in the simplest case is implemented as a one-dimensional cellular automaton executing XOR ("exclusive or") on the components of two one-dimensional bit-patterns. More precisely, if R denotes the bit-pattern associated with the receiving state of a perceptor $\not$ and S is a message pattern, then the topological complex between R and S is defined as

$$[R, S] := \text{XOR}(R, S).$$

The topological matching strength of $[R, L]$ is determined by

$$\text{top}[R, S] := (1/L(RS)) \cdot (\Sigma_i [R, L]_i),$$

where $L(RS) := \min\{L(R), L(S)\}$, $L(X)$ is the length of a bit-pattern X and $[R, S]_i$ denotes the ith components of the complex $[R, S]$. Recognition is considered successful if $\text{top}[R, S] \geqslant \Theta$, where $0 \leqslant \Theta \leqslant 1$ denotes a fixed real number.

The second evaluation tool is an adaptation mechanism. This mechanism which is effective after a recognition has been successfully accomplished and provides a set of Boolean functions responsible for the rearrangement of bits.

(4) The INTERFACE, which provides the mechanisms for modifying the "body", defining and controlling the "object stack" and the "communication module", and evaluating bit mapped color images in various ways. Much of the "interface" is menu-controlled. Important menus are:

- The **Object** menu, which provides functions for defining objects in terms of colored and labeled graphs, and grouping, ungrouping, nesting and embedding graphs into different planar topologies.
- The **Pattern** menu, which allows the user to define, randomly generate and alter one-dimensional bit-patterns.
- The **Life** menu, to assign numerical constants, variables and functions to object groups.
- The **Scanner** menu, which allows real-time user interference with the screen raster updates.
- The **Plotter** menu, which provides functions for saving and evaluating single-, or continuous-frame bitmaps with respect to particular object groups.
- The **Control** menu, to define, run, freeze and stop a simulation.

For the Macintosh Plus and II implementations of the CDM the "interface" is standardized in compliance with the APPLE "Human Interface Guidelines".

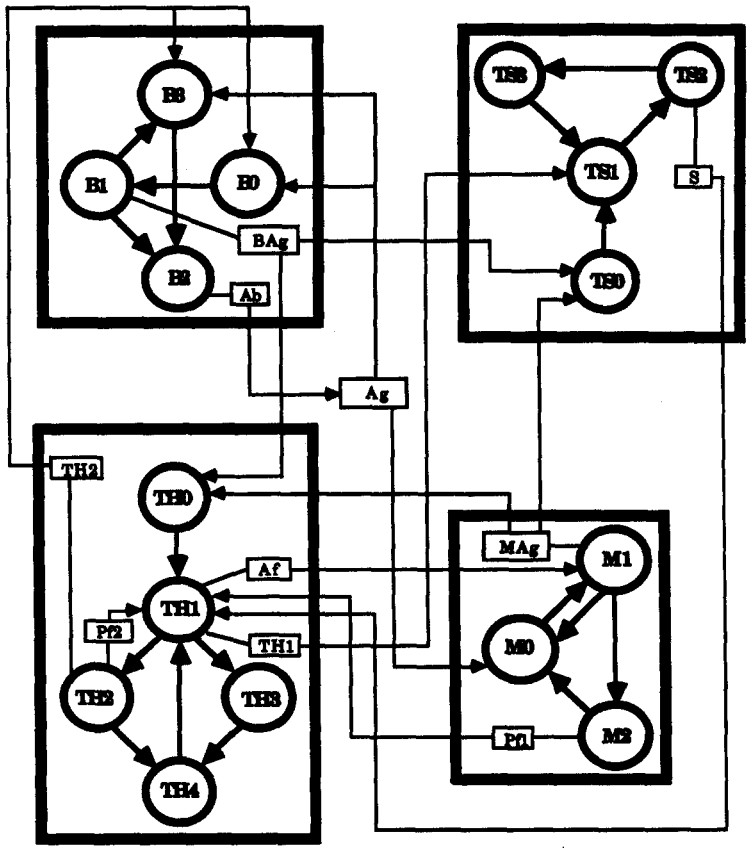
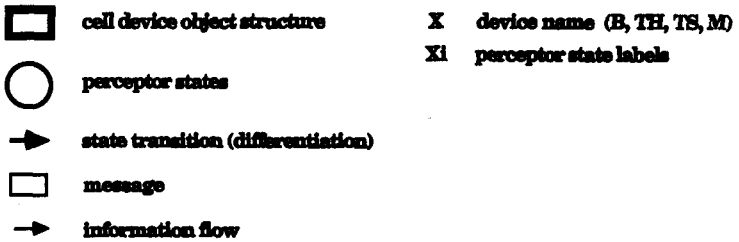


Fig. 13. Transition-graph of the BTM communications module. The figure depicts the communications module of the BTM-CDM. Drawing upon the notation introduced in Figs. 10 and 11, the following particular symbols are applied:



Antigen (Ag) is considered the driving force which, when encountered by a specific B cell device (in state B0 or B3) or a macrophage device (M0), induces a transition to state B1 (when seen in combination with a T helper cell device in state TH2) or M1, respectively. In these states, signals BAg or MAg, respectively, indicating cell/molecule complexes, are "presented". They can be recognized specifically by T helper or T suppressor cell devices (in states TH0, TS0, respectively) in a nearest neighborhood of the presenting device. In the case of MAg, cell/cell contact will induce the respective transitions to state TH1 and TS1. In state TH1, an activation factor Af (gamma-interferon) is "secreted" which can induce a neighboring M cell device to transition to state M2 (relating e.g. to microbicidal macrophage functions). Lack of Af will revert the device to state M0. In state M2, a proliferation factor Pf1 (IL-1) is secreted. Pf1 can be bound by neighboring T helper devices in state TH1, leading to a new state TH2. In this state, the T cell device can engage in cell–cell contact with a neighboring B cell device in state B0. Further, a proliferation factor Pf2 (IL-2) is released, which can induce the proliferation of T devices (TH1) in a nearest neighborhood. Lack of Pf1 will lead to the T device state TH3 thought to play a role in the activation of cytotoxic T cell devices. Both states TH2 and TH3 can be considered memory states (indicated by the symbol TH4) which can be activated again by "presentation" signals. After contact with TH0, a "presenting" B device (state B1) will transition to a plasma state B2, where specific antibody (Ab) is produced at a high rate. Otherwise, a transition into a memory state B3 will occur. B devices in this state can be activated by small amounts of the same specific Ag and will transition to B1 if the Ag signal occurs in conjunction with a TH2 signal. The suppression message S may interrupt the (TH1, TH2) or (TH1, TH3) state transition.

THE BTM—IMMUNE SYSTEM MODEL

Let us consider an antigen-driven immune system consisting of B lymphocytes, helper T lymphocytes, suppressor T lymphocytes and macrophages. Biologically, macrophages (M) bind, process and present antigen (AG) to stimulate helper T lymphocytes (TH) which then (given the right additional signals) can undergo two differentiation steps after which they are capable of stimulating B lymphocytes (B). Suppressor T lymphocytes (TS) can interrupt this process.

Using the INTERFACE functions in the **Object** menu, one can interactively and graphically construct the COMMUNICATIONS MODULE, denoted by BTM, without entering into the underlying mathematical formalism (Fig. 14). Biologically, this graph incorporates the development of the four cell types and their virtual interactions. In this context, "virtual" is used as a convenient expression to indicate dynamically allocatable activity or the "*capacity* to perform a certain action" as opposed to the "*necessity* to perform a certain action".

Mathematically, the module is a finite, nested automaton of depth 3 whose colored and labeled transition graph is represented in Fig. 13. More formally,

$$\mathrm{BTM} := ({}^{1}\Phi, {}^{1}\Sigma, {}^{1}\delta),$$

where

$${}^{1}\Phi := \{\mathrm{B}, \mathrm{TH}, \mathrm{TS}, \mathrm{M}, \mathrm{D}\},$$

$${}^{1}\Sigma := \text{the set of integers}$$

and

$${}^{1}\delta(\mathrm{X}, T) := (1 - e) \cdot \mathrm{D} + e \cdot \mathrm{X}.$$

T denotes an integer-valued function, called universal time, which evolves *independently* of BODY time. $e := \mathrm{Boole}(E(\mathrm{X}, T) \geqslant 0)$ is 1 iff the argument condition $E(\mathrm{X}, T) \geqslant 0$ is satisfied at time T and 0 otherwise. $E(\mathrm{X}, T)$ is an integer-valued function user-definable under **Life**. D denotes the empty device.

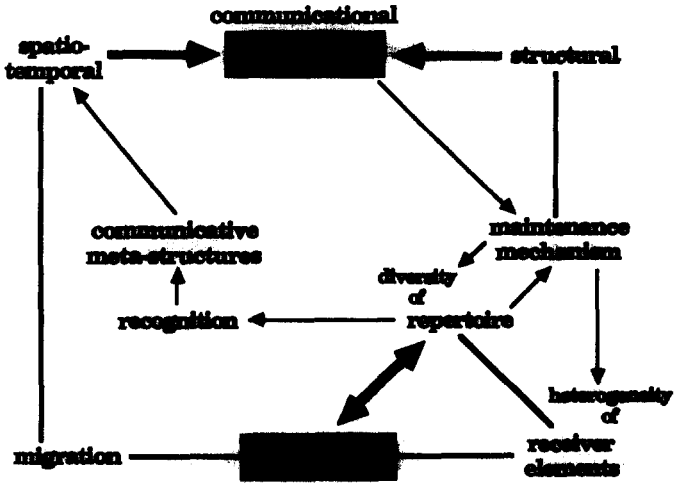


Fig. 14. The dynamics of communication in the CDM. The dynamical scenario created by a CDM simulation is influenced by: (1) three independent abstract parameters, i.e. object location, object repertoire and object structure; (2) the requirement that the system's total repertoire is diverse and (3) that diversity is maintained. Unlike most conventional dynamic system simulations, all other variables (behavior) are not part of the system definition but emerge according to a fixed algorithm capable of creating genuinely new features (communicative meta-structures). These phenomena occur locally, i.e. emerge spontaneously in a self-organized and spatially confined fashion.

The level 2 automata, i.e. cell devices, in the BTM are given as follows:

$$B_1=({}^{21}\Phi, {}^{21}\Sigma, {}^{21}\Delta, {}^{21}\lambda, {}^{21}\delta),$$

$$TH_1=({}^{22}\Phi, {}^{22}\Sigma, {}^{22}\Delta, {}^{22}\lambda, {}^{22}\delta),$$

$$TS_1=({}^{23}\Phi, {}^{23}\Sigma, {}^{23}\Delta, {}^{23}\lambda, {}^{23}\delta),$$

and

$$M_1=({}^{24}\Phi, {}^{24}\Sigma, {}^{24}\Delta, {}^{24}\lambda, {}^{24}\delta),$$

where

$${}^{21}\Phi_1=\{B01, B12, B13, B32\},$$

$${}^{22}\Phi_1=\{TH01, TH12, TH13, TH24, TH34, TH41\},$$

$${}^{23}\Phi_1=\{TS01, TS12, TS23, TS31\},$$

$${}^{24}\Phi_1=\{M01, M10, M12, M20\},$$

$${}^{21}\Sigma_1=\{AG, TH2\},$$

$${}^{22}\Sigma_1=\{BAG, MAG, PF1, S\},$$

$${}^{23}\Sigma_1=\{BAG, MAG, TH1, PF2\},$$

$${}^{24}\Sigma_1=\{AG, AF\},$$

$${}^{21}\Delta_1=\{BAG, AB\},$$

$${}^{22}\Delta_1=\{TH1, TH2, AF, PF2\},$$

$${}^{23}\Delta_1=\{S\}$$

and

$${}^{24}\Delta_1=\{MAG, PF1\}.$$

The level 3 automata, i.e. perceptors, in the case of the B cell device are

$$B01_1=({}^{311}\Phi, {}^{311}\Sigma, {}^{311}\Delta, {}^{311}\delta, {}^{311}\lambda),$$

$$B12_1=({}^{312}\Phi, {}^{312}\Sigma, {}^{312}\Delta, {}^{312}\delta, {}^{312}\lambda),$$

$$B13_1=({}^{313}\Phi, {}^{313}\Sigma, {}^{313}\Delta, {}^{313}\delta, {}^{313}\lambda),$$

and

$$B31_1=({}^{314}\Phi, {}^{314}\Sigma, {}^{314}\Delta, {}^{314}\delta, {}^{314}\lambda),$$

where

$${}^{311}\Phi_1=\{B0, B1\}, \quad {}^{311}\Sigma_1=\{AG\}, \quad {}^{311}\Delta_1=\{BAG\},$$

$${}^{312}\Phi_1=\{B1, B2\}, \quad {}^{312}\Sigma_1=\{TH2\}, \quad {}^{312}\Delta_1=\{AB\},$$

$${}^{313}\Phi_1=\{B1, B3\}, \quad {}^{313}\Sigma_1=\{TH2\}, \quad {}^{313}\Delta_1=\emptyset,$$

and

$${}^{314}\Phi_1=\{B3, B2\}, \quad {}^{314}\Sigma_1=\{AG, TH2\}, \quad {}^{314}\Delta_1=\{BAG\}.$$

The transition functions for perceptors are of the following general type:

$${}^{21}\delta(Xjk, Y)_1=(1-r)\cdot Xj + r\cdot Xk,$$

for $j, k \in \{0, \ldots, 3\}$, $X \in \{B, TH, TS, M\}$ and $r = \mathrm{rec}(Xj, Y, \Theta)$, where rec denotes the Boolean recognition function whose value, for an arbitrary input Y, is 1 if $\mathrm{top}[Xj, Y] \geq \Theta$ and 0 otherwise. As a reminder, all states, input and output labels are represented in the simulation as

one-dimensional bit-patterns some of which the user may wish to generate using the **Pattern** menu. Θ denotes a user-definable threshold value between 0 and 1. Therefore,

$$^{31i}\delta(Bj, Y) := {}^{21}\delta(Bjk, Y) = (1-r)\cdot Bj + r\cdot Bk,$$

for $1 \leqslant i \leqslant 4$. The values of the output functions are

$$^{311}\lambda(\text{B0}, Y) := \epsilon, \qquad ^{311}\lambda(\text{B1}, Y) := \text{BAG},$$

$$^{312}\lambda(\text{B1}, Y) := \epsilon, \qquad ^{312}\lambda(\text{B2}, Y) := \text{AB},$$

$$^{313}\lambda(\text{B1}, Y) := \text{BAG}, \qquad ^{313}\lambda(\text{B3}, Y) := \epsilon,$$

and

$$^{314}\lambda(\text{B3}, Y) := \epsilon, \qquad ^{314}\lambda(\text{B1}, Y) := \epsilon,$$

where ϵ denotes the empty signal, represented by an "all-blank" bit-pattern.

Given that the BODY is defined as a cellular automaton, we can think of the BTM-module occupying every site. During an actual simulation, the following restrictive conditions apply without loss of generality,

—one cell device per site is active and
—an active cell device expresses one perceptor,

at a given time. Due to these conditions, the virtual connectivity of the COMMUNICATION MODULE manifests across site boundaries leading, in real-time, to dynamically changing communication patterns.

Upon activation ("birth") a cell-device X is attributed a characteristic value called eigenlife, denoted by $E(\text{X}, T)$, which

—increases with each state transition and
—decreases with respect to universal time T.

The eigenlife $E(\text{X}, T)$ is an integer-valued function defined by restriction to the eigenlife functions of its individual perceptors:

$$E(\text{X}, T) := E(Xjk) = E(<Xjk) + \text{rec}(Xjk, Y)\cdot m(Xjk) - 1,$$

where $E(<Xjk)$ denotes the eigenlife value of the perceptor previous to Xjk. $m(Xjk)$ is a numerical identifier of Xjk, which we call its mass. During the transition from one perceptor to the next, the eigenlife is inherited from the former to the latter. If a nonpositive eigenlife value is reached, the respective cell device will discontinue, i.e. transition to a "death" state D.

APPLICATIONS

The closing section of this paper discusses projects and results which emerged while the first versions of the CDM were tested. For each application a brief presentation of the background, the objectives and the preliminary results is provided.

Application 1: regulation of lymphocyte stimulation

Background. The immune system is an anticipatory system serving as part of the host's recognition and defense mechanisms. It is hierarchically organized and consists of two major functionally overlapping compartments, termed the innate and the adaptive immune system. Both systems are integrated by a large number of cooperative processes between their cellular components. Many of these processes are initiated by antigen and evolve into complex patterns of cellular signalling through interdependent secreting and binding of immune system proteins. One of the major problems in immunology concerns the mechanisms which control the growth of such patterns.

Objectives. To determine the principles and define the functional elements which govern the regulation of growth in the immune system.

Results. As explained above, the dynamical scenario created by a CDM simulation, is influenced by: (1) three independent abstract parameters, i.e. object location, object repertoire and object structure; (2) the requirement that the system's total repertoire is diverse; and (3) that diversity is maintained for the duration of the simulation. Unlike in most conventional dynamic system simulations, all other variables (behavior) are not part of the system definition but emerge according to a fixed algorithm capable of creating genuinely new features (communicative meta-structures) (see Fig. 14). These phenomena appear locally, i.e. emerge spontaneously in a self-organized and spatially confined fashion. Any processes regulating the growth of such meta structures is revealed by their spatial confinement *for all times* and their gradual decomposition as the initiating event disappears from the scope of the system's repertoire (see Fig. 15).

The dynamics of communication in the immune system and, in particular, the regulation of proliferative growth, are special cases of this general scenario. In the first approach to the problem of growth regulation, the CDM was used in two different settings. In the first setting, a nested automaton called the (B, TH, TK, M)-module was applied to simulate B lymphocytes, T helper lymphocytes, T killer lymphocytes and macrophages and their interactions. The second setting, using the (B, TH, TK, TS, M)-module, extends the first by adding a device simulating T suppressor lymphocytes.

Two series of simulation experiments, one based on (B, TH, TK, M) and the other on (B, TH, TK, TS, M), motivated the following *a priori*:

> *Prediction.* Immune responses are self-regulated due to the combined effects of (1) local cellular networks, (2) finite cellular lifetimes and (3) the status of activating events in relation to the feed-forward mechanism at the cellular level (for details, cf. Ref. [7]). As a consequence, the regulation of proliferative growth does not depend on negative cellular regulators such as T suppressor lymphocytes.

Although we did *not* require the idiotype network hypothesis [10] to obtain the above prediction, we also did *not* as yet observe any reason to disregard networking as a regulatory *phenomenon.* In fact, our observations in this regard reveal the emergence of cooperative cellular networks as a natural consequence of immunity. In the course of a normal immune response in a healthy host system, these networks are, however, only of temporally limited stability. The duration of stability depends directly on the status of the antigenic stimulus relative to the immune system repertoire. Results explaining self–nonself discrimination in the absence of T cell suppression are contained in Ref. [11].

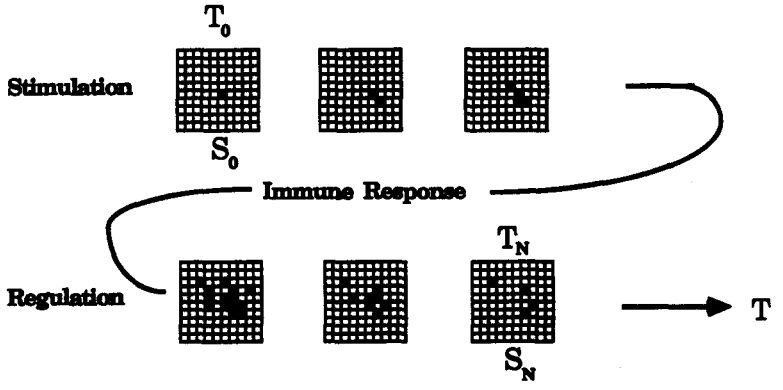


Fig. 15. Spatio-temporal evolution of a specific response. In the dynamical scenario created by a CDM, a specific response to a message evolves locally. Starting from a small group of elements capable of specifically recognizing the message, larger clusters will form due to element proliferation. Thus the response is self-amplified. Amplification does not continue indefinitely, since objects which do not receive further messages will be terminated after a while (finite eigenlife). Measurable quantities in this setup are the response duration (= TN − T0), the number and kind of objects envolved at any time point k (= Sk) and the concentrations of messages exchanged.

Application 2: simulation of disease progression in AIDS

Background. AIDS is a virally induced complex progressive disorder of the immune system with a large variety of clinical manifestations. The natural history of the infection suggests that monocytes and helper T lymphocytes are the principal cell types involved. Interactions between these cell types may generate a condition of viral persistence which ultimately leads to the collapse of immune function. Clinical evaluations and *in vitro* experimentation have been used to imply the nature of these cellular interactions but they have not yet provided a dynamic profile of the course of infection and pathogenesis. Mathematical approaches using systems of differential equations or statistical analysis [12, 13] have so far been unable to address the range and complexity caused by infection with the human immunodeficiency virus (HIV).

Objectives. To derive a formal model of the pathogenesis of AIDS *in the individual*, i.e. the uniquely determined collection of specific stages of infection (disease patterns), their connectivities (pathogenic pathways) and transition rules (accounting for disease dynamics) characteristic for AIDS.

Results. The general mathematical background on which the objectives can be achieved is the following. During the early disease stages, infection with HIV creates a basin of attraction in (TH, TK, M, B, HIV)-data space. Here, the symbols TH, TK, M and B denote helper T lymphocytes, cytotoxic T lymphocytes, macrophages and B lymphocytes, resp. During the intermittent stages of the disease, the attractor is unstable and will usually rebound periodically. Due to the peculiarities of HIV, this period of instability extends over a long time (implied latency). During this period, HIV is able to establish a reservoir of infectivity in antigen-presenting cells (APC). As the reservoir for virus in APC grows, the balance between attraction and rebound becomes critical. This (clinically) crucial point in time can be precisely determined and is mathematically described as phase-locking. A stable limit-cycle is established indicating that the disease has reached its final stages.

Depending on an individual's immune system status, the interior of this limit-cycle is characterized by various degrees of specific TH depletion, elimination of *de novo* immune responses and abundant and nonprotective antibody responses. Due to the character of immune system phase-locking as a condition contradictory to its normal dynamics (maintenance of diversity), any escape from the interior becomes impossible. We conjecture that this particularly applies to postinfection vaccination and chemotherapy, i.e. the effects of these interventions can be considered useless after the limit-cycle has been entered.

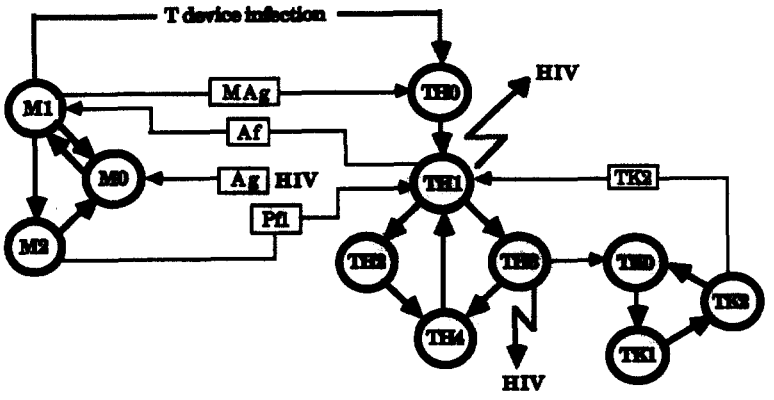


Fig. 16. The APC-virus transmission model. This figure uses the notation introduced in the BTM communication module (Fig. 13). We recall that any such diagram encodes three parameters, namely (1) the developmental structure; (2) the informational capacity of individual objects and (3) population effects originating from the composition of object populations in particular sections of the CDM body space. In the transmission model considered above, virus (HIV) is taken up by members of the (M0) macrophage population, passed along to members of the T helper population (TH0) in the course of a normal immune response (MAg), and is released as free particles upon lysis of T helpers in states TH1 or TH3, respectively. It is thought that the processes of infection and virus release are enhanced due to multiple infections (Ag distinct from HIV). Three-state T killer cell devices (TK) are considered which are activated by specific T helper devices in state TH3 to undergo a state transition from the inducible state TK0 into the state TK1 susceptible to a proliferation signal. Upon reception of this signal a proliferative transition into the cytotoxic state TK2 is performend. The latter is thought to be effective on T helper devices in state TH1.

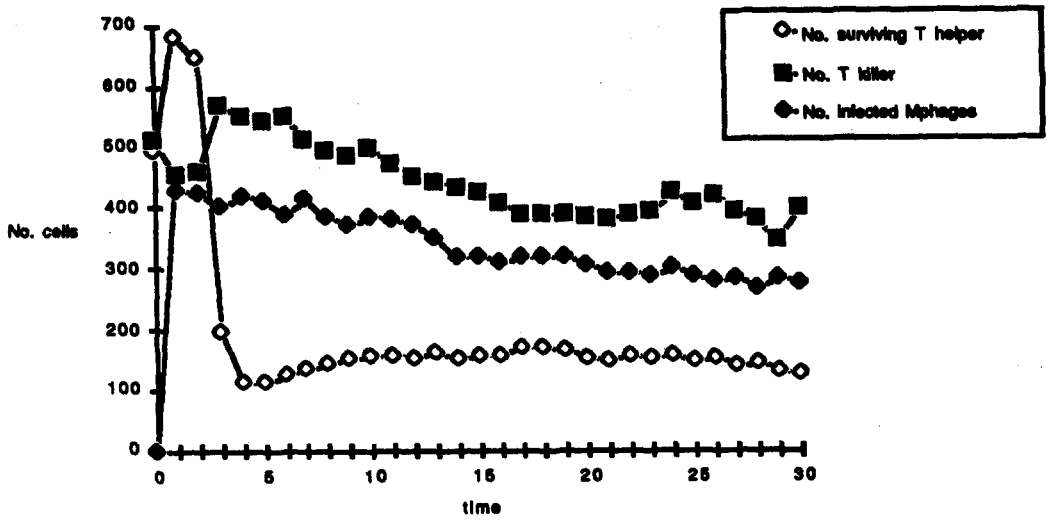


Fig. 17(a). Absolute numbers of surviving T helper cells, T killer cells and infected macrophages over time. The onset of infection is fixed at time-point $t = 0$. All cell numbers are considered with respect to the fixed volume of the simulation space.

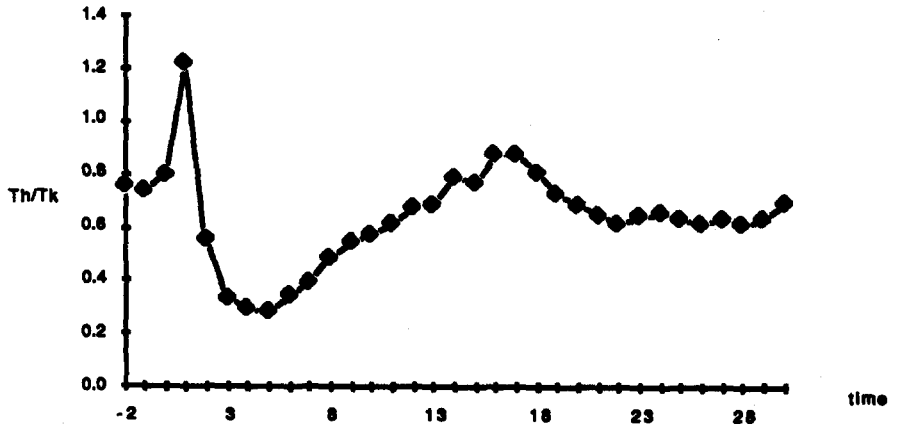


Fig. 17(b). Ratio of absolute numbers of infected T helper cells vs T killer cells. The onset of infection is fixed at time point $t = 0$.

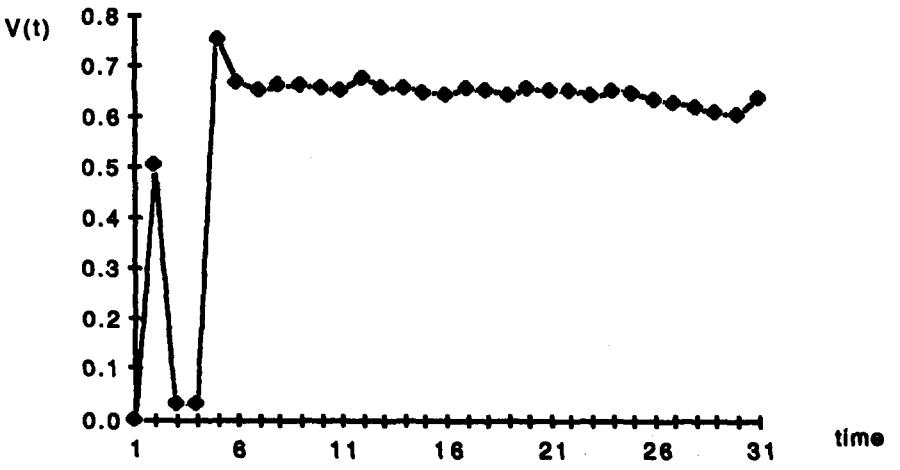


Fig. 17(c). The virus concentration $V(t)$ is defined as the number of virus particles per unit of simulation space volume. One unit can be 1024 cellular automaton sites. The onset of infection is fixed at time point $t = 1$.

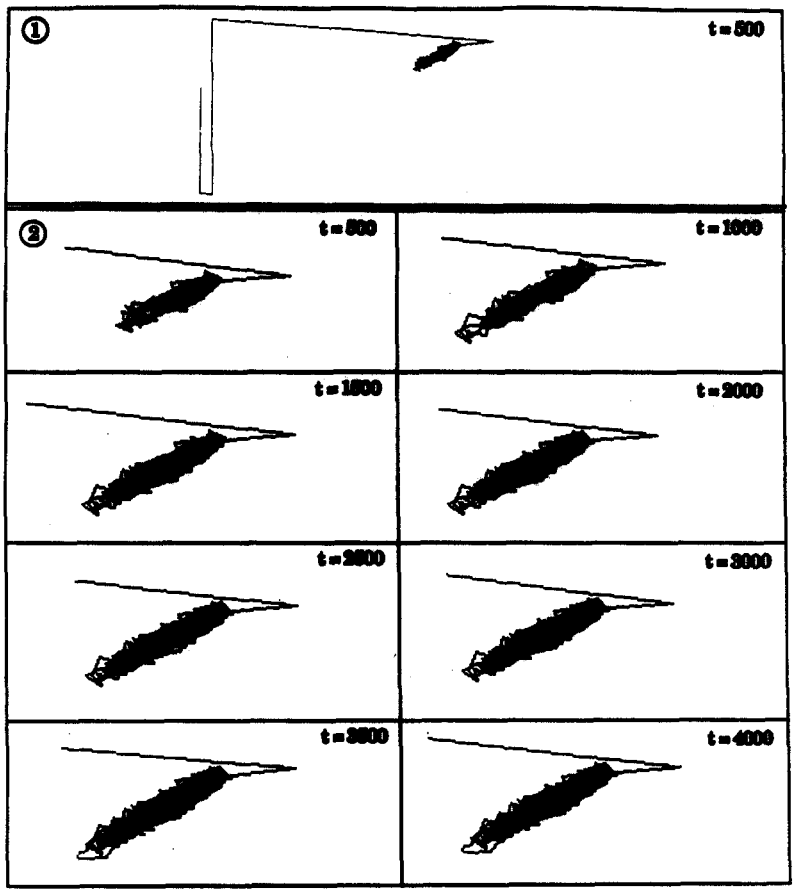


Fig. 17(d). Phase space analysis of the long-term behavior of $V(t)$. Successive discrete values of $T(t)=(V(t), V'(t))$ are connected by a line. The parameter range is $0 \leqslant V(t), V'(t) \leqslant 1$. Part (1) follows $T(t)$ from $t = 0$ (onset of infection) to $t = 500$. Part (2) consists of successive zooms onto the basin of attraction over 500 time-steps each. Here the parameter range is 0.45–0.7 in each direction.

In our first approach we considered a CDM setting using B lymphocytes (B), helper T lymphocytes (TH), cytotoxic T lymphocytes (TK) and macrophages (M) and their interactions characteristic for a normal immune response and implemented the virus transmission model shown in Fig. 16. A simulation experiment using the APC-model and a maximum of 10^4 cells per cell type of varying specificities, shows the population dynamical effects, upon and after virus introduction, in Fig. 17. Figure 17(a) indicates in absolute numbers the surviving specific TH cells, the TK cells and the infected M cells over dimensionless time. Factors can be derived [14–16] (2.25 ± 0.75 in the case of TH) to scale these numbers and the unit of time to available clinical data. Figure 17(b) depicts the ratio of surviving TH cells vs TK cells over time. The evolution of virus concentration over time, denoted $V(t)$, is shown in Fig. 17(c). The figures are based on the data obtained during the first 30 time-steps of the simulation experiment. Figure 17(d) provides a trend-analysis of the long-term behavior of $V(t)$ over several thousand discrete time-steps. A basin of attraction in $(V(t), V'(t))$-phase space is established, consisting of several islands which are visited periodically. Here, $V' := dV/dt$. This reveals that the APC-model indeed accounts for the establishment of a virus reservoir. This process is supported by the population of an APC type (M) during the course of normal immune system functions.

Acknowledgements—I would like to express my thanks to Dr Melvin Cohn for many valuable suggestions and counter-arguments. I am glad to acknowledge the invaluable help and the many patiently suggested clarifications I received from Drs Christa Müller-Sieburg and C. David Pauza in numerous discussions about the immune system and its diseases. Furthermore, I am indebted to Drs Ronald Schwartz, Klaus Rajewski and Arnold Mandell for substantial critisisms and many valuable suggestions. I am very grateful to Dr Jim Ostlund from the Biocomputing Laboratory at the Salk Institute for his interest and active support of the work reported in this paper.

Finally, I would like to thank the immune system for providing the pleasure associated with the creation, implementation and exploitation of novel concepts.

REFERENCES

1. P. Davies, The creative cosmos. *New Scient.* **17,** 41 (1987).
2. G. McCalla and N. Cercone, Techniques and issues in the design of applied artificial intelligence systems. *Computers Math. Applic.* **11,** 421 (1985).
3. A. Yonesawa and M. Tokoro, *Object-oriented Concurrent Programming.* MIT Press, Cambridge, Mass. (1987).
4. J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Reading Mass. (1979).
5. Wolfram St., *Theory and Applications of Cellular Automata. Advanced Series on Complex Systems,* Vol. 1. World Scientific, Singapore (1986).
6. H. B. Sieburg and C. E. Müller-Sieburg, Cellular device machines: an approach to mathematical modeling of infectious diseases. *Nucl. Phys. B (Proc. Suppl.)* **2,** 615 (1987).
7. H. B. Sieburg, A logical dynamic systems approach to the regulation of antigen-driven lymphocyte stimulation. In *Theoretical Immunology, SFI Series in the Sciences of Complexity* (Ed. A. S. Perelson). Addison-Wesley, Reading, Mass. (1988).
8. J. R. Lump, The value of theoretical models in immunological research. *Immun. Tod.* **4,** 209 (1983).
9. M. Jilek and D. Prikrylova, Some notes on mathematical modeling of the immune response. In *Immunology and Epidemiology* (Eds G. W. Hoffmann and T. Hraba). *Lecture Notes in Biomathematics,* No. 65. Springer, Berlin (1986).
10. N. K. Jerne, Towards a network theory of the immune system. *Ann. Immun. (Inst. Pasteur)* **125C,** 373 (1974).
11. R. J. de Boer and P. Hogeweg, Self–nonself discrimination due to immunological nonlinearities: the analysis of a series of models of numerical methods. *IMA Jl Math. appl. Med. Biol.* **4,** 1 (1987).
12. L. N. Cooper, Theory of an immune system retrovirus. *Proc. natn. Acad. Sci. U.S.A.* **83,** 9159 (1986).
13. R. M. Anderson, R. M. May, G. F. Medley and A. M. Johnson, A preliminary study of the transmission dynamics of the human immunodeficiency virus (HIV). *Science* (in press).
14. R. R. Redfield, D. C. Wright and E. C. Tramont, The Walter Reed staging classification for HTLV-III/LAV infection. *N. Engl. J. Med.* **314,** 131 (1986).
15. B. S. Dobozin *et al.*, The relationship of abnormalities of cellular immunity to antibodies to HTLV-III in homosexual men. *Cell. Immun.* **98,** 156 (1986).
16. W. Lang *et al.*, Clinical, immunologic, and serologic findings in men at risk for acquired immunodeficiency syndrome. *JAMA* **257,** 326 (1987).
17. B. Bollobas, *Graph Theory.* Springer, New York (1979).
18. Ph. E. Seiden and L. S. Schulman, Percolation and galaxies. *Science* **233,** 425 (1986).
19. W. D. Hillis, *The Connection Machine.* MIT Press, Cambridge Mass. (1985).
20. J. D. Murray, E. A. Stanley and D. L. Brown, On the spatial spread of rabies among foxes. Technical Report 86-0479, Los Alamos Natn. Lab., N.M. (1986).
21. E. Jen, Invariant strings and pattern-recognizing properties of one-dimensional cellular automata. *J. statist. Phys.* **43,** 243 (1986).
22. T. Toffoli, Cellular automata mechanics. Ph.D., Computer Science Dept., Univ. of Michigan, Ann Arbor (1977).
23. T. Toffoli and N. Margolus, *Cellular Automata Machines.* MIT Press, Cambridge, Mass. (1987).

APPENDIX A

Graphs

For further reading on this subject we recommend Ref. [17] and the literature cited therein.

Definition. A graph Γ consists of a set $N(\Gamma)$, a set $C(\Gamma)$ and two maps

$$C(\Gamma) \to N(\Gamma) \times N(\Gamma); \qquad d \mapsto (o(d), t(d))$$

and

$$C(\Gamma) \to C(\Gamma); \qquad d \mapsto i(d),$$

which satisfies the following conditions: for each $d \in C(\Gamma)$, we have

$$d = i(i(d)), \quad d \neq i(d) \quad \text{and} \quad o(d) = t(i(d)).$$

The elements of $N(\Gamma)$ and $C(\Gamma)$ are called the nodes and the connections of Γ, resp. The extremities $o(d)$ and $t(d)$ are called the origin and the terminus of a connection d, resp.

An orientation of a graph Γ is a subset $C_+(\Gamma)$ of $C(\Gamma)$, such that

$$C(\Gamma) = C_+(\Gamma) \cup i(C_+(\Gamma)).$$

If such a subset exists then the graph is called oriented.

Since the general definition of a graph does not impose any restrictions as to the particular character of its node and connection elements, it is possible to build graphs from graphs:

Definition. A graph Γ is called (finitely) nested iff there exists an integer $k \geqslant 1$ and graphs $\Gamma_1, \ldots, \Gamma_k$, such that

$$N(\Gamma) = \{N(\Gamma_1), \ldots, N(\Gamma_k)\}$$

and

$$C(\Gamma) = \{d | \exists n, m \leqslant k: o(d) \in N(\Gamma) \wedge t(d) \in N(\Gamma)\}.$$

The nesting process can be repeated to arrive at hierarchical and heterarchical organizations in graphs which, as we will show later, are useful tools for the modeling of structurally complex systems. The notion of depth in such graphs is derived in an obvious manner.

Sequential orderings or paths, i.e. rules describing the sequence in which particular connections of a graph are accessible, can be defined by using the oriented Path_n-graph given for an arbitrary integer $n \geqslant 0$ as follows:

$$N(\text{Path}_n) := \{0, 1, \ldots, n\}$$

and

$$C(\text{Path}_n) := \{[i, i+1]: 0 \leqslant i \leqslant n-1, o([i, i+1]) = i, t([i, i+1]) = i+1\},$$

where $[a, b]$ denotes the section of the real line between a and b, $a \leqslant b$. Then

Definition. A path of length n in a graph Γ is a morphism of Path_n into Γ, i.e. there exist mappings

$$N(\text{Path}_n) \to N(\Gamma)$$

and

$$C(\text{Path}_n) \to C(\Gamma).$$

The section is concluded with a list of important additional properties which can be associated with a graph

Definition. (1) A graph is said to be connected if any two nodes are the extremities of at least one path. The maximal connected subgraphs (under the relation of inclusion) are called the connected components of the graph.

(2) A graph is called finite iff it has a finite number of nodes and connections. It is called locally finite iff every node is the extremity of a finite number of connections. It is called labeled when certain attributes or symbols are associated with its nodes and colored when certain attributes are associated with its connections.

To abbreviate nomenclature, the term blockdiagram will be used to indicate a directed, colored and labeled graph.

APPENDIX B

Automata

Automata (with output) are mathematical structures capable of converting any number of messages which are compatible with their input capacity into output messages. During the conversion, a set of (transition) rules is applied to link a discrete set of (automaton) states to the input messages.

In their applications, automata—in particular, cellular automata—have been shown to be flexible and efficient tools for the computer simulation of complicated dynamical processes. Such processes may arise in areas as diverse as fluid dynamics [5], the evolution of galaxies [18], parallel distributed processing and computation [19], epidemiology [20] and pattern recognition [21]. Systems whose behavior is characterized by complicated process involving the interaction of many components of nonuniform structure are called complex systems. The role of applied automaton theory in mathematical modeling is to provide a simple, realistic, computer- and user-digestible approach to the simulation and, ultimately, the understanding of such systems.

More formally (for further information, cf. Ref. [4]),

Definition. A finite automaton with output A is a quintuple $(\Phi, \Sigma, \Delta, \delta, \lambda)$ consisting of a finite set of states Φ, a finite input alphabet Σ, a finite output alphabet Δ and two transition functions

$$\delta: \Phi \times \Sigma \to \Phi; \qquad q, s) \to p = \delta(q, s)$$

and

$$\lambda: \Phi \to \Delta; \qquad q \to s' = \lambda(q).$$

Using the terminology introduced in the preceding section, we can associate a directed graph Γ_A, called the transition diagram, with a finite automaton with output (FAO) A as follows:

$$N(\Gamma_A) = \Phi$$

and

$$C(\Gamma_A) = \{d | \exists q \in \Phi: \exists s \in \Sigma: o(d) = q \wedge t(d) = \delta(q, s)\}.$$

Thus, by construction, the transition diagram of an automaton is labeled and colored. To simplify notation, one can denote the connections d by the input symbol s responsible for the transition. Further, the symbols A and Γ_A can be applied interchangeably.

Finite automata can be connected in various ways to yield larger and/or more sophisticated automata. Although very little of this has so far been rigorously explored, it is known that already the simplest of these connection schemes yield powerful parallel computers. For example, a cellular automaton can be thought of as an infinite collection of identical finite

automata whose state sets are coupled locally. The automata are conveniently placed into the fundamental domains of a tessellated finite dimensional topological space and each is then connected to a fixed set of immediately neighboring automata. This set is called the neighborhood template. Since the tessellation provides for identical neighborhoods, the connection rule is uniformly valid for all templates. In summary,

> *Definition.* A cellular automaton (CA) is a quadruple (A, G, U, f), where A denotes a finite automaton, G a tesselation group of a topological space H and U a neighborhood template. f is a map from A^n into A, where n denotes the number of fundamental domains in U. The rank of G is called the dimension of the cellular automaton.

For more information on CA refer to Refs [5, 22, 23]. The BTM-machine provides an example for a more complex connection scheme.

APPENDIX C

Object-oriented Programming

The term "object" emerged independently in various fields in computer science, almost simultaneous in the early 1970s, to refer to notions that were different in their appearence, yet mutually related [3]. All of these notions were invented to manage the complexity of software systems in such a way that objects represent components of a modularly decomposed system (modular units of knowledge representation). Typical notions of object that have emerged in various fields are:

- computational agents that carry out their actions in response to a message, such as Hewitt's *actors*, in parallel computation models and AI programming;
- packages of information with *class/instance* and *super-class/sub-class* hierarchies such as objects in Simula and Smalltalk, in simulation, AI and more general programming;
- *abstract data types* in programming languages;
- modules or units for knowledge and expertise, such as Minsky's *frame*, in knowledge representation; and
- protected *resources* in operating systems.

The common characteristics of these notions are that

> *Definition.* An object is a logical or physical entity that is *self-contained* and provided with a *unified communication protocol.*

In simple words, an object—on the computer—consists of a set of data and a piece of code capable of interpreting the data and advising the object what to do with them.

Computation or information processing in the object-oriented framework is represented as a sequence of message-passing among objects. Since an object is a self-contained entity provided with a unified communication protocol, the decomposition of a system into a collection of objects is very flexible and the resulting modular system structure becomes very natural. Furthermore, unified communication protocols and "self-containedness' protect objects from illegitimate access, thus supporting orderly interactions among objects. These advantages provide the almost perfect ground for concurrent, i.e. parallel, programming.

Object-oriented concurrent programming (OCP) is a powerful programming and design methodology (style), in which the system to be constructed is represented (modeled) as a collection of *concurrently* executable objects and the interactions among the system components are represented by message-passing. The object orientation and metaphors of parallel problem-solving strategies that are found in many complex biological systems identify OCP as the ideal agent for the design and implementation of computation models of such systems. Conversely, such metaphors provide a basis for inventing more sophisticated problem-solving schemes that exploit parallelism. Mathematically, any computation model developed under OCP represents a dynamical system. This is largely due to the flexibility of the notions of objects which, in addition to their common characteristics of self-containedness and unified communication protocols, allow for properties such as development, growth, recognition, innovation and adaptation. An object may be activated by a message arrival and become inactive after a series of actions, or it may always be active and receive messages by executing certain commands. Different kinds of intraobject actions may be allowed and be characterized in either an imperative, functional or logical style. More than one chain of action may or may not take place within an object simultaneously. Some actions may be expressed in terms other than message-passing.